

# **Analizador de manos de NLHE siguiendo estrategias configurables**

**Ángel Álvarez Muñoz**

**Pablo Conca Gran**

**Héctor Jesús Figueroa Cadenas**

**GRADO EN INGENIERÍA DE COMPUTADORES**  
**UNIVERSIDAD COMPLUTENSE DE MADRID**



**TRABAJO FIN DE GRADO**

**06/06/2014**

Directores: Alberto Núñez Covarrubias / Manuel Núñez García



## Autorización de difusión y utilización

Nosotros, Ángel Álvarez Muñoz, Pablo Conca Gran y Héctor Jesús Figueroa Cadenas, autores del presente documento y del proyecto *Analizador de manos de NLHE siguiendo estrategias configurables*, autorizamos a la Universidad Complutense de Madrid a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la propia memoria, código los contenidos audiovisuales y/o el prototipo desarrollado.

Ángel Álvarez Muñoz  
DNI: 11858546E

Pablo Conca Gran  
DNI: 11863787L

Héctor Jesús Figueroa Cadenas  
DNI: 76133114D



## **Agradecimientos**

En primer lugar queremos dar las gracias a nuestros directores Manuel Núñez García y Alberto Núñez Covarrubias por su ayuda y paciencia al resolvernos dudas.

También queremos agradecerse a la comunidad de EducaPoker, por darnos permiso para utilizar sus documentos y por probar la aplicación y darnos feedback sobre ella.

A Álvaro Serrano González, por ayudarnos con unos diseños tan profesionales.

A nuestras familias, por el apoyo que nos han dado a lo largo de estos años, por lo que hoy este trabajo es una realidad.

Y por último, a nuestros amigos y parejas, por soportarnos y darnos algún consejo que otro cuando lo necesitábamos.

A todos vosotros, gracias.



## RESUMEN

En esta memoria se describe el Trabajo de Fin de Grado "Analizador de manos de NLHE siguiendo estrategias configurables", que consiste en la creación de una herramienta que ayude a jugadores de "Texas hold'em" póker en la modalidad "cash" en mesas de seis jugadores a analizar sus jugadas una vez terminada la partida.

Estas jugadas se analizan en función de unas estrategias creadas por la escuela de póker online EducaPoker<sup>1</sup>. Los usuarios de la aplicación podrán introducir archivos con los registros de sus manos y comprobar si han jugado siguiendo la estrategia, o si por el contrario han cometido algún error.

En caso de haberlo cometido se mostrará el punto de la partida en el que se ha producido, y se verá también el movimiento que deberían haber efectuado para seguir la estrategia correctamente.

Como los jugadores pueden no querer adaptar exactamente su juego a las estrategias predefinidas, la aplicación ofrece la opción al usuario de crear las suyas propias, modificando las estrategias básicas de EducaPoker.

Como parte de este Trabajo de Fin de Grado, también se ha creado un sitio Web<sup>2</sup> para que los usuarios puedan conocer la aplicación. En ella se explica su funcionamiento, los cambios efectuados en las distintas versiones, y cómo descargarla y ejecutarla. Además, la Web ha sido creada para hacer un seguimiento del uso de la aplicación.

**Palabras clave:** Póker, estrategias, analizador, Java.

---

<sup>1</sup>[www.educapoker.com](http://www.educapoker.com)

<sup>2</sup>[www.pokercashanalyzer.co.nf](http://www.pokercashanalyzer.co.nf)

## ABSTRACT

In this memory, we describe our project, which consists in a tool that helps poker players to analyze their moves once the game is ended.

These moves are analyzed following some strategies created by the online poker school “EducaPoker<sup>3</sup>”. The users of this application can load files with their hand logs in order to check if they have played by following a strategy. Also this application checks if users have made any mistake, seeing exactly the point of the game where they played wrong, and also seeing which action they should have done.

Due to players may not want to adapt their playstyle to the provided strategies, the application gives the user the choice to create new strategies by modifying the basic strategies of EducaPoker.

Also as part of this project, we have created a website<sup>4</sup> in order to show our application to the users. In this website, we explain how it works, we show the changelog, and we explain how to download and run it.

The website also helps us to track the use of the application.

**Keywords:** Poker, strategies, analyzer, Java.

---

<sup>3</sup>[www.educapoker.com](http://www.educapoker.com)

<sup>4</sup>[www.pokercashanalyzer.co.nf](http://www.pokercashanalyzer.co.nf)



## Índice general

Índice general .....	1
<b>1. ÁMBITO DE LA APLICACIÓN .....</b>	<b>3</b>
<b>2. TECNOLOGÍA UTILIZADA.....</b>	<b>5</b>
2.1 Lenguajes y librerías .....	5
2.2 Entornos de desarrollo utilizados .....	6
2.3 Control de versiones.....	6
2.4 Pruebas .....	6
<b>3. PLANIFICACIÓN .....</b>	<b>7</b>
3.1 Modelo de desarrollo.....	7
3.2 Inconvenientes.....	8
3.3 Trabajo en equipo.....	8
3.4 Reuniones y comunicación.....	8
3.5 Gestión de código y archivos .....	8
<b>4. ESTRUCTURA DE LA APLICACIÓN .....</b>	<b>11</b>
4.1 Descripción de los paquetes .....	11
4.2 Diagrama UML .....	12
<b>5. DESCRIPCIÓN DE LOS BLOQUES DE LA APLICACIÓN .....</b>	<b>13</b>
5.1 Vistas.....	13
5.1.1 Vista inicial .....	13
5.1.2 Vista de la mesa de juego .....	17
5.2 Parsers .....	18
5.2.1 Funcionamiento.....	18
5.2.2 Palabras clave .....	19
5.2.3 Fases del juego .....	20
5.2.4 Otros datos recogidos .....	20
5.2.5 Ejemplo de una mano.....	21
5.3 Anonimizador.....	22
5.4 Estrategias .....	22
5.4.1 Introducción .....	22
5.4.2 Descripción de estrategias.....	22

5.4.3 Implementación .....	30
<b>6. PARTE SOCIAL .....</b>	<b>31</b>
<b>7. GLOSARIO .....</b>	<b>35</b>
<b>8. CONCLUSIONES Y TRABAJO FUTURO .....</b>	<b>37</b>
8.1 Conclusiones .....	37
8.2 Trabajo futuro.....	37
<b>9. CONCLUSIONS AND FUTURE WORK .....</b>	<b>39</b>
9.1 Conclusions .....	39
9.2 Future work .....	39
<b>10. BIBLIOGRAFÍA.....</b>	<b>41</b>



## **1. ÁMBITO DE LA APLICACIÓN**

En los últimos años, el póker online ha aumentado su número de jugadores de manera muy significativa. Es por ello que el nivel de los jugadores aumenta, y también el nivel de estudio del juego por parte de éstos.

A raíz de ello, han surgido numerosos autores y escuelas que tratan de enseñar las bases matemáticas del juego, así como de desarrollar estrategias que ayuden a los jugadores a mejorar su estilo de juego.

Este proyecto gira en torno a las estrategias de una de estas escuelas, llamada EducaPoker, conocida por ser la primera escuela creada por jugadores de habla hispana.

El funcionamiento de esta escuela consiste en asociar la cuenta de EducaPoker del jugador a su cuenta en alguna de las principales casas de póker online, de manera que a medida que va jugando, se va incrementando el número de puntos de su cuenta, lo que le permite subir niveles. Según vaya subiendo de nivel, el jugador tendrá acceso a artículos y estrategias cada vez más avanzadas.

Para esta aplicación, se ha utilizado información de artículos y estrategias pertenecientes a los niveles más bajos. Las estrategias básicas que se han implementado son: EP35, EP50 y EP100.

Los números del título de las estrategias definen el número de ciegas en la partida. Cada estrategia se divide en dos fases principales: PreFlop y PostFlop.

La primera fase se divide en seis situaciones, que dependen de cómo lleguen las apuestas al jugador. Cada una de estas etapas es tratada con una tabla que contiene todas las combinaciones de las cartas y varios colores que determinan la posición del jugador en la mesa. Esto se relaciona de esta forma: si las cartas que el jugador posee tienen el color (en la tabla) de la posición en la que él mismo se encuentra, debe hacer lo que la tabla de la estrategia estima. Entre las estrategias cambian las posiciones desde las que se realizan las apuestas, así como la cantidad de las mismas.

La segunda fase se divide en tres etapas: Flop, Turn y River. Estas fases se analizan en tres etapas que son definidas dependiendo de cómo haya sido jugada la fase de PreFlop. A parte de esto, también se hace distinción de si se juega contra un rival o contra varios en cada momento. Toda la fase de PostFlop se basa en la fuerza de la mano del jugador principal, del número de Outs que tiene y de cómo hayan jugado el rival o los rivales. Se diferencian unas estrategias de otras en las restricciones (cada vez más complejas) para realizar las apuestas.

En cuanto a las casas de póker, el programa acepta manos tanto de PokerStars como de 888poker. Nos decidimos por estas ya que son dos de las casas más importantes y con mayor número de jugadores a nivel mundial. En cualquier caso, la herramienta está preparada para que introducir un bloque de procesamiento de manos de otra casa sea relativamente sencillo.

Respecto a quién puede utilizar la aplicación, al hacerla se ha pensado en algo dirigido tanto a jugadores que se estén iniciando en el estudio del póker, como a usuarios más experimentados. Por un lado, las estrategias básicas implementadas son relativamente simples, pero por otro, se da la opción a usuarios más avanzados de editar dichas estrategias para crear una que se adapte más a su estilo de juego.



## 2. TECNOLOGÍA UTILIZADA

En este capítulo se describe la tecnología utilizada para el desarrollo de la aplicación<sup>5</sup>.

### 2.1 Lenguajes y librerías

- **Java:** lenguaje de programación de propósito general, concurrente, orientado a objetos y basado en clases que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo, lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. Java es, a partir de 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones de cliente-servidor de web.

Se ha desarrollado la aplicación en Java, ya que posee librerías útiles para el desarrollo de la misma, además de que se puede encontrar más documentación de Java que de cualquier otro lenguaje.

Dentro de Java, cabe nombrar las principales librerías que hemos utilizado. Para generar las diferentes vistas de la aplicación hemos utilizado las librerías AWT y Swing, y para mandar correos a la dirección de contacto hemos utilizado la librería JavaMail.

- **AWT:** AbstractWindowToolkit. Kit de herramientas de gráficos, interfaz de usuario, y sistema de ventanas independiente de la plataforma original de Java.
- **Swing:** biblioteca gráfica para Java. Incluye widgets para interfaz gráfica de usuario tales como cajas de texto, botones, desplegados y tablas.
- **JavaMail:** expansión de Java que facilita el envío y recepción de e-mail desde código java. JavaMail no se incluye en la JDK ni en la JRE, sino que debe conseguirse como un paquete externo.
- **HTML5 (HyperTextMarkupLanguage, versión 5):** quinta revisión importante del lenguaje básico de la World Wide Web, HTML. Todavía se encuentra en modo

---

<sup>5</sup>Las definiciones están sacadas de es.wikipedia.org

experimental, lo cual indica la misma W3C, aunque ya es usado por múltiples desarrolladores web por sus avances, mejoras y ventajas.

- **CSS:** lenguaje de hojas de estilo utilizado para describir el aspecto y el formato de un documento escrito en un lenguaje de marcas.
- **Bootstrap:** framework o conjunto de herramientas de software libre para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como, extensiones de JavaScript opcionales adicionales.

## 2.2 Entornos de desarrollo utilizados

- **Eclipse:** programa informático compuesto por un conjunto de herramientas de programación de código abierto típicamente usado para desarrollar entornos de desarrollo integrados (del inglés IDE), como el IDE de Java llamado Java Development Toolkit (JDT).

Hemos utilizado Eclipse en su última versión estable (Kepler), ya que es el entorno al que más familiarizados estamos, además, al ser el más popular, incluye numerosas extensiones muy útiles.

## 2.3 Control de versiones

- **Git:** es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente.

Para llevar un control de versiones del código, se ha creado un repositorio gestionado con Git. Se decidió utilizar esta tecnología ya que ofrece más opciones que otras herramientas similares, además, se creyó que era bueno que se aprendiera a usarlo.

## 2.4 Pruebas

- **JUnit:** es un conjunto de clases (framework) que permite realizar la ejecución de clases Java de manera controlada, para poder evaluar si el funcionamiento de cada uno de los métodos de la clase se comporta como se espera.

Dentro del proyecto, se ha incluido un paquete con clases de prueba que se utilizan para testear los nuevos métodos que se van creando. Se ha utilizado JUnit por su sencillez y porque viene integrado en Eclipse.

### 3. PLANIFICACIÓN

En este capítulo se describe cómo ha sido el proceso de planificación, diseño, implementación y pruebas a lo largo de los meses que ha durado el proyecto.

#### 3.1 Modelo de desarrollo

El proyecto se ha desarrollado a lo largo de varias fases, durante todo el curso.

Se ha utilizado un modelo de desarrollo en cascada, es decir, siguiendo unas fases muy bien definidas y ordenadas. A continuación se describen dichas fases, que pueden ver representadas en el diagrama de Gantt de las figuras 3.1 y 3.2.

- **Definición del proyecto (01/Oct/2013 - 18/Oct/2013):** fase de puesta en marcha del proyecto. En un principio, el tutor expone la idea del proyecto. A continuación, se realiza una investigación acerca de software similar al que se pretende crear, barajando varias ideas. Después de eso, se van puliendo los objetivos a cubrir en sucesivas reuniones con el tutor, hasta que finalmente se llega al acuerdo de qué es exactamente lo que se quiere crear.
- **Análisis y especificación de requisitos (18/Oct/2013 - 04/Nov/2013):** fase donde se junta la información obtenida tanto en las reuniones con el tutor como de investigar otras aplicaciones. Se obtienen los requisitos de la aplicación. Una vez hecho esto, se determina el comportamiento que debe tener.
- **Diseño (04/Nov/2013 - 22/Nov/2013):** una vez que se sabe clara la funcionalidad de la aplicación y sus requisitos, se entra en esta fase, en la que se define la estructura de datos y clases que va a tener el programa, y cómo se van a comunicar entre sí.
- **Implementación (22/Nov/2013 - 23/May/2014):** en esta etapa se desarrolla el código que representa el diseño pensado en la fase anterior. Esta etapa también incluye un análisis de las herramientas y lenguajes de desarrollo para decidir cuál de ellas se adecua más a las necesidades del proyecto.
- **Pruebas:** se han realizado pruebas tanto durante la implementación como después de ésta. Para las pruebas a nivel de código, se ha incluido un paquete “Test” en la aplicación para agrupar todos los test de JUnit que se han pasado. Para las pruebas a nivel de funcionamiento de la aplicación, además de las ya mencionadas, se ha subido la

aplicación a EducaPoker para que los propios usuarios la prueben y así obtener valoraciones externas.

- **Documentación:** en esta fase se genera la documentación correspondiente al desarrollo de la aplicación. Dicha documentación se va haciendo al mismo tiempo que la fase de implementación, ya que es la forma más rigurosa y eficiente.
- **Redacción de la memoria (12/May/2013 - 06/Jun/2014):** última fase del proceso. Comienza a finales de la implementación, sobre las partes que ya están finalizadas y probadas, para que los resultados expuestos en ella sean definitivos.

### 3.2 Inconvenientes

Aunque las fases estaban bien definidas, así como el trabajo a realizar en cada una, ha habido una serie de inconvenientes a la hora de seguir esta planificación, debido a cambios en los requisitos provocados por el cambio de alguna de las estrategias del propio EducaPoker.

### 3.3 Trabajo en equipo

A la hora de trabajar, no hemos tenido ningún jefe de proyecto, ya que éramos un grupo reducido, por lo que el reparto de tareas se ha hecho de forma descentralizada y lo más equitativamente posible, en algunos casos haciendo algunas entre varios miembros del grupo.

### 3.4 Reuniones y comunicación

Entre los miembros del grupo se han hecho reuniones regularmente, normalmente cada semana o cada dos semanas, tanto presenciales como por Skype. En ellas se han discutido las tareas que pendientes y se ha repartido trabajo para hacer hasta la siguiente reunión.

Las reuniones con los tutores han sido, en un principio, para concretar los detalles del proyecto, y más tarde han sido reuniones de seguimiento, para aclarar conceptos y para fijar plazos. La mayoría de las reuniones han sido presenciales pero también se han hecho reuniones utilizando Hangouts.

La comunicación entre miembros del grupo y tutores ha sido, además de mediante las reuniones ya mencionadas, principalmente por Gmail.

### 3.5 Gestión de código y archivos

Se ha alojado el código en el sitio web Bitbucket, ya que permite crear un repositorio privado y accesible a todos los miembros del equipo<sup>6</sup>.

---

<sup>6</sup><https://bitbucket.org/>



Para gestionar las diferentes versiones de la aplicación, como se ha comentado antes, se ha utilizado Git, debido a que tiene más opciones que otros programas similares y es muy sencillo de integrar en Eclipse.

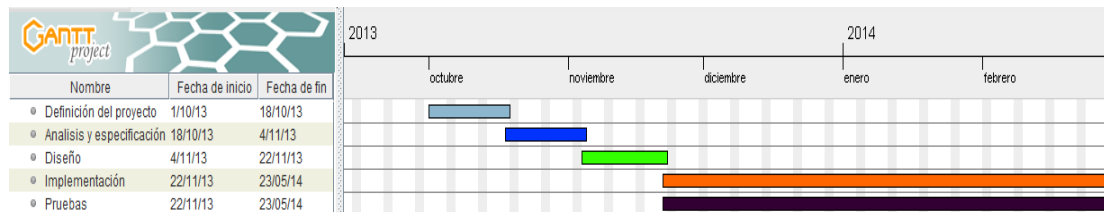


Figura 1.1 – Diagrama de Gantt para los meses de octubre a febrero

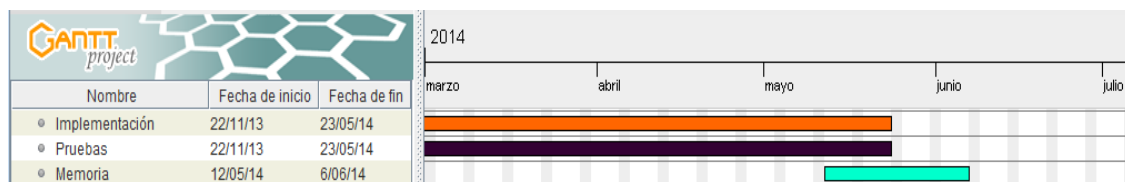


Figura 1.2 – Diagrama de Gantt para los meses de marzo a junio





## 4. ESTRUCTURA DE LA APLICACIÓN

En este capítulo se describe la estructura del proyecto a nivel de código, en primer lugar los paquetes de los que está compuesto, y por último el diagrama UML (Figura 4.1) donde se puede ver gráficamente la distribución de dichos paquetes.

### 4.1 Descripción de los paquetes

La implementación de la aplicación se va a dividir en los siguientes paquetes:

- **Controlador:** módulo controlador que comunica al modelo con la vista y le da a esta última los datos para que los muestre.
- **Estrategias:** módulo que contiene la lógica de las estrategias. Están incluidas tanto las estructuras necesarias para almacenar las estrategias como los métodos para analizar jugadas en función de esas estrategias.
- **Juego:** módulo donde están las clases que representan una partida de póker (cartas, mesa, jugadores, etc).
- **Main:** módulo que contiene la clase principal para poder lanzar la aplicación.
- **Modelo:** módulo donde se almacena el estado de los objetos durante una ejecución.
- **Parsers:** módulo donde se encuentran las clases que permiten analizar archivos de manos y obtener datos de ellos, y así poder tratarlos en las estrategias.
- **Resources:** módulo donde están los recursos necesarios para la aplicación, como imágenes.
- **Test:** módulo dedicado a hacer pruebas sobre diferentes partes de la aplicación.
- **Util:** módulo donde se encuentran clases con funciones de utilidad común a toda la aplicación.
- **Vista:** módulo donde están definidas las interfaces gráficas para que los datos puedan ser visualizados.

## 4.2 Diagrama UML

En el siguiente diagrama (figura 4.1) se ve la representación gráfica de la estructura de paquetes que sigue la aplicación

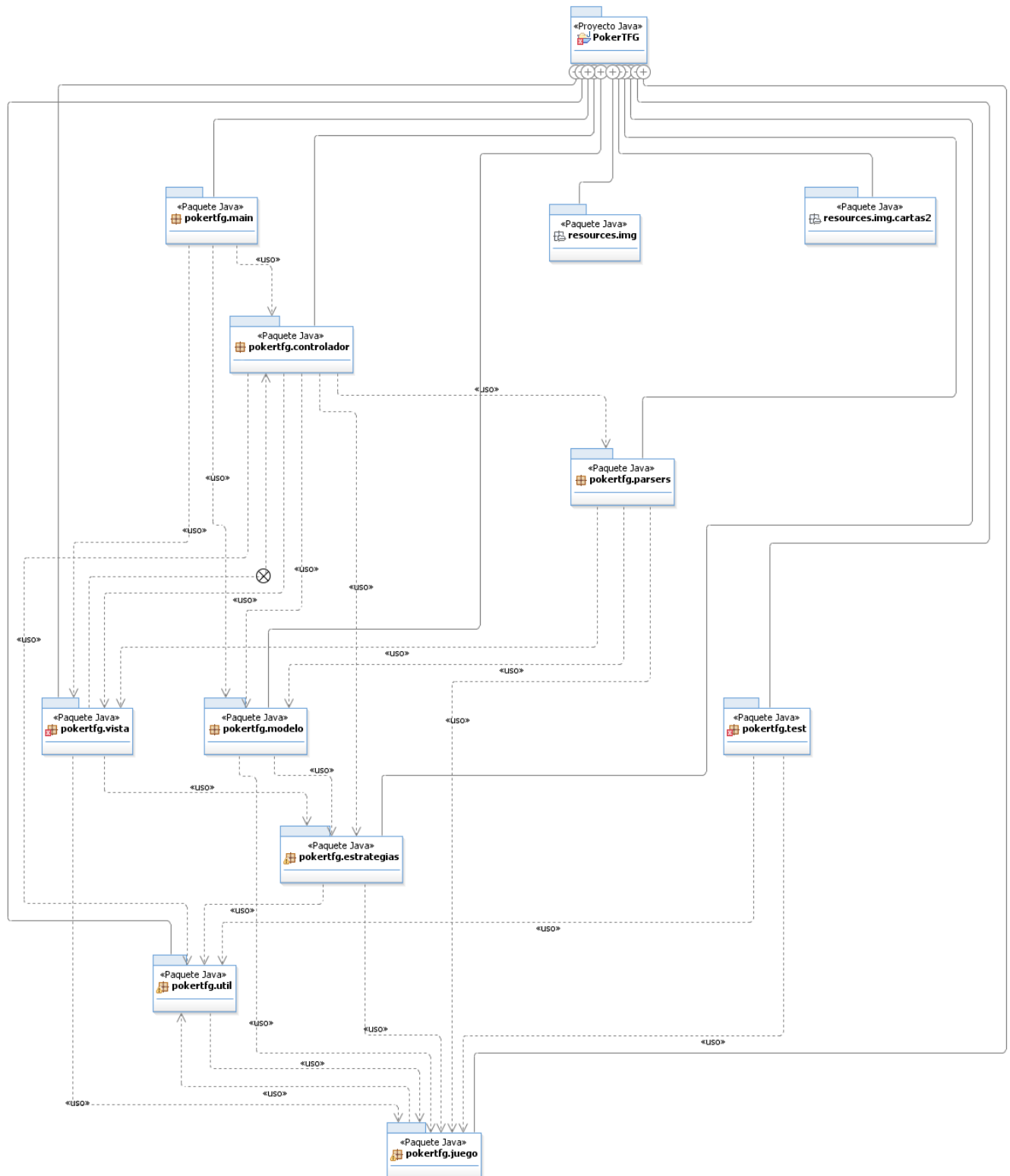


Figura 4.1 – Representación UML de la estructura del proyecto



## 5. DESCRIPCIÓN DE LOS BLOQUES DE LA APLICACIÓN

En este capítulo se describe cada uno de los bloques de la aplicación a nivel de programación.

### 5.1 Vistas

La aplicación cuenta con dos vistas. Este parte del trabajo ha sido coordinada por Pablo Conca Gran. A continuación se detalla cada una de ellas, tanto sus componentes como su implementación:

#### 5.1.1 Vista inicial

Es la vista encargada de recoger los parámetros de la aplicación (la mano a procesar, la estrategia y la casa) y de mostrar la tabla de la estrategia seleccionada.

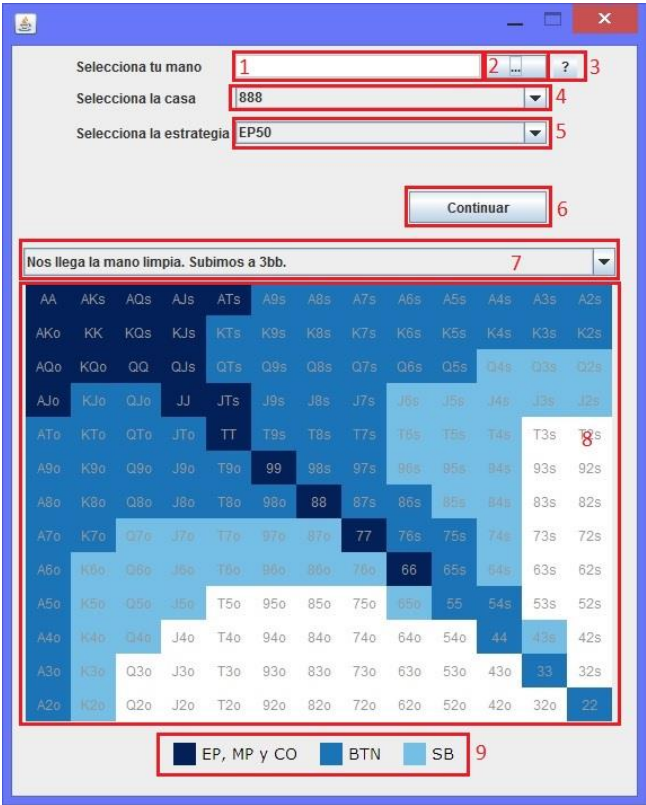


Figura 5.1 – Vista principal de la aplicación

Esta vista (figura 5.1) tiene los siguientes componentes:

1. Cuadro de texto donde se muestra la ruta del archivo de manos.
2. Selector del archivo de manos. Se abrirá una ventana que nos permitirá explorar las carpetas para seleccionar el archivo de manos (figura 5.2).

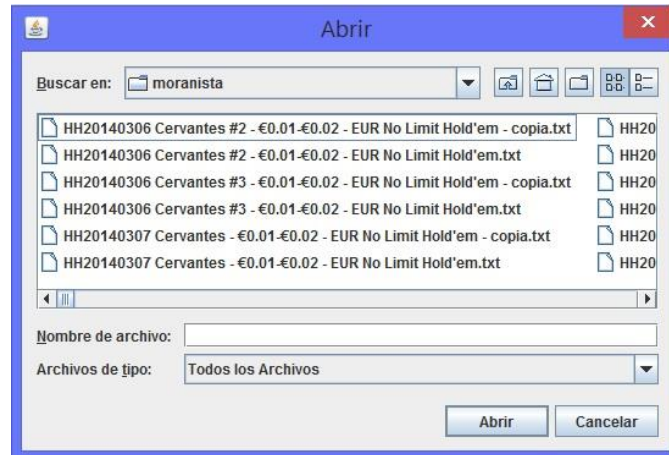


Figura 5.2 – Ventana de selección de archivos

3. Botón para contactar con el desarrollador. Al pulsarlo se abrirá una ventana (figura 5.3) en la cual podremos rellenar dos campos que serán el asunto y el contenido, respectivamente, de un correo que nos llegará a una dirección que nos creamos para dar soporte a los usuarios de la aplicación.

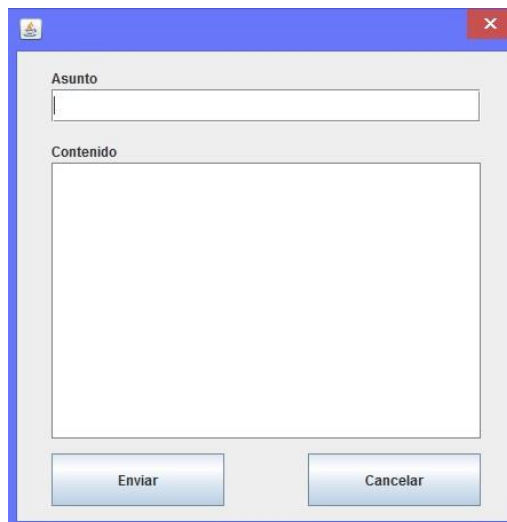


Figura 5.3 – Ventana de contacto con el desarrollador

4. Selector de casa. Sirve para elegir la casa de las manos a procesar. Las opciones son “888poker” y “PokerStars”.

5. Selector de la estrategia. Sirve para elegir la estrategia en función de la cual va a evaluar las manos que entran. Las opciones son “EP35”, “EP50”, “EP100” y “Personalizada”. Las tres primeras opciones son estrategias predefinidas, mientras que la última sirve para activar las estrategias personalizadas, que muestra dos nuevos botones (cargar y guardar estrategia) y desbloquea la tabla de estrategias para que sea editable (figura 5.4).
6. Botón de continuar. Sirve para, una vez completados todos los campos, realizar la validación de la mano y pasar a la vista de la mesa.
7. Selector de la situación. Sirve para elegir la situación que tiene que producirse para que se aplique la estrategia de la tabla. Cada una de las opciones de este selector tiene una tabla asociada. Estas situaciones varían según la estrategia, y son las siguientes:

Para EP35<sup>7</sup>:

- Nos llega la mano limpia. Subimos a 4bb<sup>8</sup>.
- ROL (Raise Over Limpers). Subimos a 4bb.
- Nos llega el bote subido. Resubimos a 8bb.
- Enfrentándonos a una cuarta apuesta. Vamos All-in.
- Enfrentándonos a una resubida. Vamos All-in.
- Bote subido y resubido. Vamos All-in.

Para EP50:

- Nos llega la mano limpia. Subimos a 3bb.
- ROL (Raise Over Limpers). Subimos a 5bb.
- El bote nos llega subido. Subimos a 9bb.
- Nos enfrentamos a una cuarta apuesta. Vamos All-in.
- Cuarta apuesta. Vamos All-in.
- Bote subido y resubido. Vamos All-in.

Para EP100:

- Nos llega la mano limpia. Subimos a 4bb.
- ROL (Raise Over Limpers). Subimos a 4bb.
- El bote nos llega subido. Subimos a 9bb.
- Nos enfrentamos a una cuarta apuesta. Vamos All-in.
- Cuarta apuesta. Vamos All-in.

---

<sup>7</sup>Nombre de la estrategia: EP se refiere a EducaPoker y la cifra al número de ciegas.

<sup>8</sup>Big blind o ciega grande.

8. Tabla de estrategias. Nos muestra las tablas pertenecientes a la estrategia que haya marcada en el selector. Si la estrategia seleccionada es alguna de las predefinidas, la tabla simplemente mostrará la estrategia, pero si la seleccionada es la personalizada, la tabla será editable al pulsar sobre ella (figura 5.4).
9. Leyenda. Indica el significado de los colores de la tabla, que pueden cambiar según la estrategia seleccionada.

Vista que se muestra al seleccionar la estrategia personalizada:

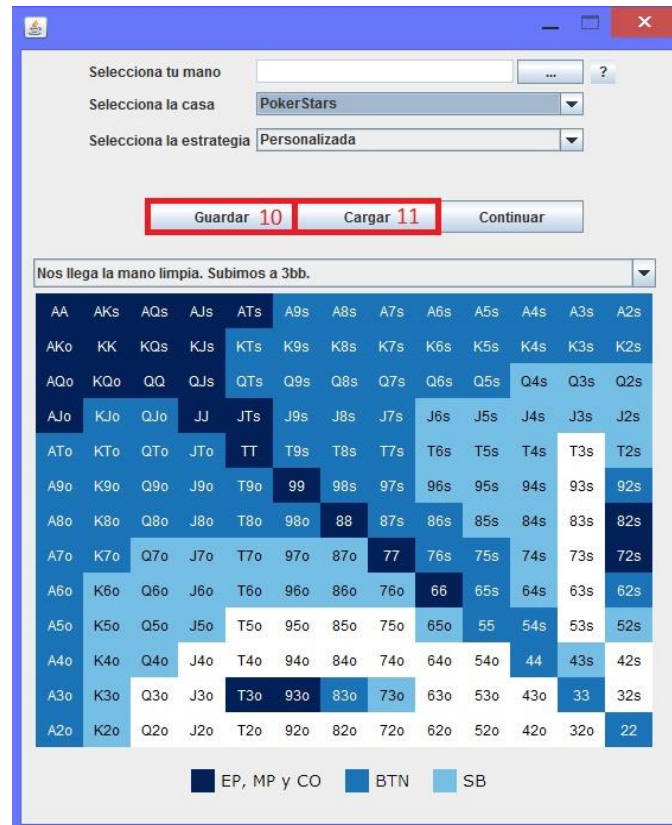
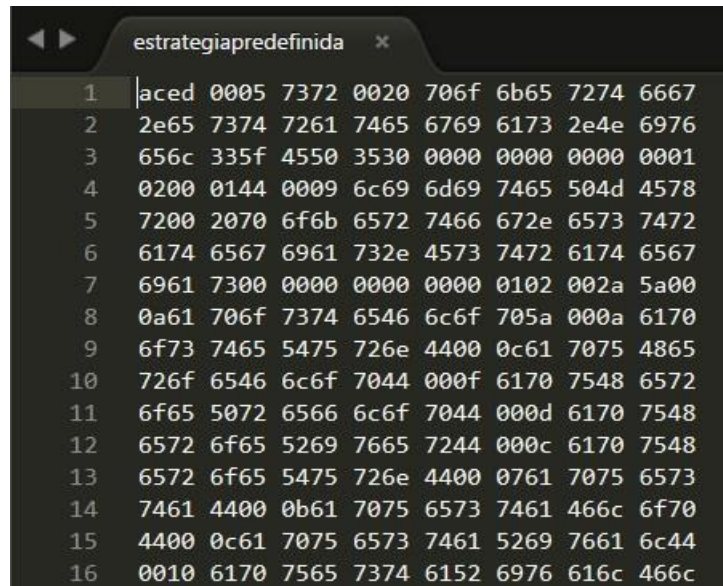


Figura 5.4 – Vista principal al seleccionar estrategia personalizada

10. Botón de guardar. Sirve para generar un archivo con la información almacenada en ese momento en las tablas de estrategias. Guardamos el objeto java con todas sus tablas directamente en el archivo, por lo que dicho archivo tiene el aspecto que se muestra en la figura 5.5.
11. Botón de cargar. Sirve para cargar una estrategia que hayamos guardado previamente.





	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	aced	0005	7372	0020	706f	6b65	7274	6667								
2	2e65	7374	7261	7465	6769	6173	2e4e	6976								
3	656c	335f	4550	3530	0000	0000	0000	0001								
4	0200	0144	0009	6c69	6d69	7465	504d	4578								
5	7200	2070	6f6b	6572	7466	672e	6573	7472								
6	6174	6567	6961	732e	4573	7472	6174	6567								
7	6961	7300	0000	0000	0000	0102	002a	5a00								
8	0a61	706f	7374	6546	6c6f	705a	000a	6170								
9	6f73	7465	5475	726e	4400	0c61	7075	4865								
10	726f	6546	6c6f	7044	000f	6170	7548	6572								
11	6f65	5072	6566	6c6f	7044	000d	6170	7548								
12	6572	6f65	5269	7665	7244	000c	6170	7548								
13	6572	6f65	5475	726e	4400	0761	7075	6573								
14	7461	4400	0b61	7075	6573	7461	466c	6f70								
15	4400	0c61	7075	6573	7461	5269	7661	6c44								
16	0010	6170	7565	7374	6152	6976	616c	466c								

Figura 5.5 – Archivo de estrategia generado por la aplicación

### 5.1.2 Vista de la mesa de juego

Es la vista encargada de representar las manos procesadas, donde se puede ver cada fase de cada una de las manos introducidas.



Figura 5.6 – Vista de la mesa de juego

Esta vista (figura 5.6) tiene los siguientes componentes:

1. Barra lateral. Nos muestra una pestaña por cada mano que haya procesado del archivo. Si pinchamos en alguna de las manos, se abrirán una serie de paneles correspondientes a las fases que haya habido en esa mano. Al pinchar en cada uno de los paneles, podremos ver las cartas que teníamos y las cartas que había en la mesa en ese momento de la partida.

Los símbolos que aparecen a la derecha del panel de fases representan si las jugadas han seguido las estrategias que hemos indicado en la ventana anterior o no:

- El “tick” verde significa que el movimiento realizado es el correcto siguiendo la estrategia.
  - La cruz roja significa que el movimiento realizado es incorrecto según la estrategia.
  - La cruz amarilla significa que el movimiento realizado es correcto pero las cantidades apostadas no lo son.
2. Cartas del jugador. Nos muestra las cartas repartidas al jugador en la mano seleccionada.
  3. Cartas en la mesa. Según la fase de cada mano sobre la que pulsemos, se nos muestran las cartas que hay en la mesa en ese momento.
  4. Información sobre la estrategia. Nos muestra el movimiento correcto según la estrategia, y el movimiento que hemos realizado, por lo que podemos ver en qué nos hemos equivocado.  
También nos muestra, si se da el caso, la cantidad apostada y la cantidad que deberíamos haber apostado según la estrategia seleccionada.
  5. Ficha de dealer. Nos muestra qué jugador ocupa la posición de BTN en cada mano.

## 5.2 Parsers

Durante el desarrollo del proyecto, hemos implementado dos parsers de manos. Estos parsers permiten recoger información de las casa 888poker y PokerStars. Este parte del trabajo ha sido coordinada por Ángel Álvarez Muñoz.

### 5.2.1 Funcionamiento

El parser irá recorriendo línea a línea el archivo y buscará mediante palabras clave los datos que queremos recoger. Los iremos almacenando en una variable mesa del tipo mesaJuego, que contiene lo siguiente:

- **Un array de jugadores** para almacenar los jugadores y sus apuestas, nombres, movimientos, etc.
- **MaxApu y minApu:** variables para almacenar la ciega grande y la ciega pequeña respectivamente.
- **Bote:** variable para almacenar el bote acumulado por las apuestas de los jugadores.
- **Un array de fase** para ir almacenando los datos según la fase en la que estemos y poder diferenciarlas después de una manera sencilla.
- **Un array de turno** para saber en qué orden actúa cada jugador.

- **Array de cartas (flop y partida):** array donde se almacenan las cartas que van saliendo en la mesa.
- **Arrays de movimientos buenos:** arrays donde se guardan los movimientos que habría que realizar según la estrategia.
- **Arrays de movimientos reales:** arrays donde se guardan los movimientos o jugadas que ha hecho el jugador y que son recogidos por el parser.
- **ApuestaHeroe:** array para almacenar las apuestas del héroe recogida por el parser.
- **ApuestaEstrategia:** array para almacenar qué apuestas deberían haberse realizado según la estrategia.
- **NumFase:** contador con el que iremos cambiando de fase y distinguiremos entre una fase y otra.

Estas "mesas" las guardaremos en un array de manos. Dicho array será pasado al modelo para tratar los datos recogidos.

### 5.2.2 Palabras clave

El parser funciona buscando palabras clave dentro de las manos para extraer los datos a través de ellas. Estas palabras clave son las siguientes:

- **PokerStars:** con ella sabemos que estamos en la primera línea, luego buscaremos el primer paréntesis de la línea y de ahí sacaremos los valores de la ciega grande y la ciega pequeña.
- **Button:** De aquí obtendremos que posición de la mesa es el button.
- **Seat:** En estas líneas obtendremos el nombre del jugador, el número de su asiento y el dinero que tienen. También comprobaremos que si el asiento coincide con el del button, se le asignará a ese jugador la posición de button.
- **Posts:** Aquí sabremos que jugadores son la ciega pequeña y la ciega grande (en este orden).
- **Dealt:** De esta línea obtendremos el nombre del héroe y sus cartas. Si en la mano no existe la línea con esta palabra, es que el héroe no está jugando.
- **Folds, bets, raises, all-in, checks:** veremos qué jugador ha hecho que acción. Iremos sumando al bote las cantidades apostadas para posteriores cálculos.
- **Leaves:** si algún jugador abandona la mesa, lo eliminaremos del array de Jugadores.
- **Collect:** para determinar que jugador ha ganado y cuanto dinero se lleva.
- **FLOP, TURN, RIVER:** de estas líneas iremos obteniendo las cartas que van saliendo en la mesa.
- **SUMMARY:** aquí es donde diremos que deje de procesar datos de esa mano y la guarde si está jugando el héroe.

### 5.2.3 Fases del juego

Durante la lectura de la mano, se irán almacenando las diversas fases del juego en una variable de tipo EstadoJuego, que está en la variable de la mano.

En un principio se empieza automáticamente en preFlop, y luego mediante variables booleanas y al leer la línea correspondiente del archivo, se irá cambiando de fase. Al llegar a la línea de SUMMARY se reiniciarán las variables necesarias.

Todas las manos se irán almacenando en un array de manos que luego se usará para ir sacando las manos y obteniendo los datos para mostrarlos.

### 5.2.4 Otros datos recogidos

Durante la recolección de datos se puede ver qué jugadores ganan las manos y cuándo apuestan o suben. Esto lo utilizamos para calcular el VPIP.

VPIP o VP\$IP indica el dinero voluntariamente introducido en el bote y señala en qué porcentaje de casos un jugador pone dinero en el bote antes del flop. Si un jugador que está en la ciega ve la flop sin inversión adicional porque nadie sube, entonces no cuenta para el cálculo de esta estadística.

Según este dato, podemos evaluar a los jugadores de la siguiente manera:

<b>Nit</b>	Si el VPIP es menor de 16%.
<b>Tight_preflop</b>	Si el VPIP está entre el 16% y el 22%.
<b>Semiloose_preflop</b>	Si el VPIP entre el 23% y el 30%.
<b>Loose_preflop</b>	VPIP entre el 31% y el 40%.
<b>Fish</b>	VPIP mayor que el 40%.

### 5.2.5 Ejemplo de una mano

A continuación podemos ver el formato de uno de los archivos que la aplicación puede analizar<sup>9</sup>:

```
PokerStars Zoom Hand #112877034697: Hold'em No Limit (€0.01/€0.02) -
2014/03/07 0:02:37 CET [2014/03/06 18:02:37 ET]
Table 'Cervantes' 6-max Seat #1 isthebutton
Seat 1: Anon1 (€1.04 in chips)
Seat 2: Heroe (€1 in chips)
Seat 3: Anon2 (€1 in chips)
Seat 4: Anon3 (€4.35 in chips)
Seat 5: Anon4 (€2.14 in chips)
Seat 6: Anon5 (€1.48 in chips)
Heroe: postssmallblind €0.01
Anon2: postsbigblind €0.02
*** HOLE CARDS ***
DealttoHeroe [As 6c]
Anon3: folds
Anon4: folds
Anon5: folds
Anon1: raises €0.06 to €0.08
Heroe: folds
Anon2: calls €0.06
*** FLOP *** [6d 3s Jc]
Anon2: checks
Anon1: bets €0.08
Anon2: calls €0.08
*** TURN *** [6d 3s Jc] [Kh]
Anon2: checks
Anon1: bets €0.24
Anon2: calls €0.24
*** RIVER *** [6d 3s JcKh] [8c]
Anon2: checks
Anon1: bets €0.64 and isall-in
Anon2: calls €0.60 and isall-in
Uncalledbet (€0.04) returnedto Anon1
*** SHOW DOWN ***
Anon1: shows [Kc Qc] (a pair of Kings)
Anon2: shows [KsQs] (a pair of Kings)
Anon2 collected €0.97 frompot
Anon1 collected €0.96 frompot
*** SUMMARY ***
Total pot €2.01 | Rake €0.08
Board [6d 3s JcKh 8c]
Seat 1: Anon1 (button) showed [Kc Qc] and won (€0.96) with a pair of Kings
Seat 2: Heroe (smallblind) foldedbeforeFlop
Seat 3: Anon2 (bigblind) showed [KsQs] and won (€0.97) with a pair of Kings
Seat 4: Anon3 foldedbeforeFlop (didn'tbet)
Seat 5: Anon4 foldedbeforeFlop (didn'tbet)
Seat 6: Anon5 foldedbeforeFlop (didn'tbet)
```

<sup>9</sup>Los nombres de los jugadores han sido anonimizados previamente.

### 5.3 Anonimizador

Al pedir archivos de manos a la gente, nos dimos cuenta de que mucha gente prefiere que sus datos y estadísticas no fuesen visibles, aun siendo una aplicación offline. Debido a esto se ha implementado una pequeña funcionalidad aparte que permite extraer todos los nombres de los jugadores y sustituirlos por nombres como, por ejemplo, "Anon1", excepto al propio usuario de la aplicación, que lo podremos encontrar como "Héroe" dentro de la partida.

El funcionamiento de este anonimizador es similar al del parser, solo que no recoge datos, sólo nombres, y de una forma ordenada los sustituye por su correspondiente "ANON".

### 5.4 Estrategias

A continuación se va a describir el funcionamiento de las estrategias utilizadas, así como su implementación.

#### 5.4.1 Introducción

La aplicación analiza tres estrategias de la escuela de póker EducaPoker. Estas estrategias son: PreFlopIII y PostFlopIII de EP35, EP50, EP100, siendo dichos números el número de ciegas. El número tres de las estrategias define el nivel, que en este caso es de dificultad "Universidad". Este proyecto también contiene una sección de estrategias configurables. Este parte del trabajo ha sido coordinada por Héctor J. Figueroa Cadenas.

#### 5.4.2 Descripción de estrategias

En esta sección se describirá cómo jugar con las tres estrategias implementadas en la aplicación, tanto la fase PreFlop como la fase PostFlop.

##### 5.4.2.1 EP35

A continuación se va a describir cómo funciona la estrategia EP35.

- **PreFlop:** en esta estrategia se empieza a introducir conceptos que justifican nuevos movimientos. Estos conceptos son la Defensa de Ciegas y el Raise Over Limpers (subir cuando hay jugadores que igualaron la ciega grande).

La estrategia aplica diferentes tácticas. Se resubirá con más manos cuando alguien suba desde la posición botón o desde la ciega pequeña, y se irá All-in con más facilidad cuando se esté en zona de robo y algún jugador resuba.

A continuación, se detalla cómo jugar según las diferentes situaciones en la que se encuentre el jugador.

- Llega la mano limpia

Si la mano llega limpia al jugador (nadie ha apostado antes que él), habría que realizar una subida a 4bb (cuatro veces el precio de la ciega grande), según la tabla de la estrategia en este caso.

Como se muestra en la siguiente tabla (Figura 5.7<sup>10 11</sup>), en la leyenda se asocia un color a la posición de la mesa (Ej.: azul con tono claro a la posición de la mesa *SB*, small blind: ciega pequeña). El jugador debe apostar lo que dice la estrategia si la mano del mismo coincide con los valores de la tabla que están en rango de apuesta.

Nos llega la mano limpia. Subimos a 4bb.

AA	AKs	AQs	AJs	ATs	A9s	A8s	A7s	A6s	A5s	A4s	A3s	A2s
AKo	KK	KQs	KJs	KTs	K9s	K8s	K7s	K6s	K5s	K4s	K3s	K2s
AQo	KQo	QQ	QJs	QTs	Q9s	Q8s	Q7s	Q6s	Q5s	Q4s	Q3s	Q2s
AJo	KJo	QJo	JJ	JTs	J9s	J8s	J7s	J6s	J5s	J4s	J3s	J2s
ATo	KTo	QTo	JTo	TT	T9s	T8s	T7s	T6s	T5s	T4s	T3s	T2s
A9o	K9o	Q9o	J9o	T9o	99	98s	97s	96s	95s	94s	93s	92s
A8o	K8o	Q8o	J8o	T8o	98o	88	87s	86s	85s	84s	83s	82s
A7o	K7o	Q7o	J7o	T7o	97o	87o	77	76s	75s	74s	73s	72s
A6o	K6o	Q6o	J6o	T6o	96o	86o	76o	66	65s	64s	63s	62s
A5o	K5o	Q5o	J5o	T5o	95o	85o	75o	65o	55	54s	53s	52s
A4o	K4o	Q4o	J4o	T4o	94o	84o	74o	64o	54o	44	43s	42s
A3o	K3o	Q3o	J3o	T3o	93o	83o	73o	63o	53o	43o	33	32s
A2o	K2o	Q2o	J2o	T2o	92o	82o	72o	62o	52o	42o	32o	22

■ EP, MP y CO ■ BTN ■ SB

Figura 5.7

- ROL (Raise Over Limpers)

Cuando llegue el turno de actuar y haya algún jugador que haya igualado la ciega grande, se subirá a 4 ciegas, independientemente del número de limperts (jugadores que han igualado la ciega grande).

Para saber cuándo hacer esta apuesta, basta con mirar la tabla correspondiente a esta situación en esta estrategia.

<sup>10</sup>Nota: Los colores oscuros envuelven a los más claros, es decir, si las cartas tienen el color de *BTN* (button) o *EP* (early position), *MP* (middle position) o *CO* (cut-off position) y el jugador se encuentra en la posición *SB* (que está definido con un color más claro), sería una combinación con la que habría que apostar. Por ejemplo, si se tiene en la mano una pareja de reyes (KK) y se está en la posición *SB*, está en rango de apuesta.

<sup>11</sup>La “s” que hay después del valor de las cartas en la tabla significa que son del mismo palo, y la “o” que son de distinto palo.

- Llega el bote subido

En el caso de que el bote llegue subido al jugador, siguiendo las indicaciones de la tabla asociada a esta situación, hay que realizar una resubida de 8bb (ocho veces el precio de la ciega grande).

- Enfrentándose a una cuarta apuesta

Si llega una resubida tras una apuesta y dos subidas posteriores, se irá con All-in siguiendo la tabla de la estrategia.

- Enfrentándose a una resubida

Si un rival realiza una apuesta y otro hace una resubida, se irá con All-in siguiendo la tabla de la estrategia.

- Bote subido y resubido

En esta situación, como en las dos anteriores, se irá con All-in siguiendo la tabla de la estrategia.

- **PostFlop:** en este nivel se introducen algunos conceptos muy importantes y relevantes para todas las estrategias. Estos conceptos son: la evaluación de la fuerza de la mano y calidad de los outs.

La fase PostFlop se divide en tres: juego cuando hemos subido preflop, cuando hemos resubido preflop y cuando hemos pasado en la ciega grande. Dentro de cada una se analizará las fases Flop, Turn y River. Para la decisión de las apuestas en todas las fases, hay que tener en cuenta los movimientos del rival, la fuerza de la mano y el número de outs.

- Cuando se ha subido PreFlop

La mayor parte de manos que juguemos PostFlop entrarán en este apartado, ya que cuando jugamos una mano PreFlop voluntariamente, lo hacemos subiendo.

Flop: cuando se apueste en el Flop, el tamaño de la apuesta será igual al bote que hay en la mesa. Se distinguen dos formas de jugar, una es contra un solo rival, y la otra contra varios rivales.

Turn: en esta fase se diferencian en dos formas de jugar, contra un solo rival o contra varios.

River: en el river, casi nunca se va a tener que tomar una decisión, porque ya estaremos All-in la mayor parte de las veces.



No obstante, es posible llegar al River sin estar All-in. Esto sucederá principalmente porque en el Turn no se tenía una mano lo suficientemente fuerte para apostar, pero que debido al PME, podamos igualar una apuesta de nuestro oponente.

Sin embargo, las manos débiles (fuerza de la mano) aún pueden ganar la mano. Muchas veces el rival pasará y de esas algunas se ganará la mano. Además, cuando el rival apuesta, es posible que lo haga con un farol.

- Juego cuando se ha resubido PreFlop

Cuando se resune a 8bb (ocho veces el precio de la ciega grande), el bote en el flop ya es muy grande, y con una mano débil se querrá llegar al showdown. El tamaño de apuesta cuando se resube será de medio bote. Hay distinciones al jugar contra un solo rival o contra varios.

- Juego cuando se ha pasado en la ciega grande

Nuestra apuesta aquí siempre será de tamaño del bote.

Flop: se apostará el bote siempre que tengamos al menos una mano media (fuerza de la mano) o con un proyecto de al menos 8 outs. Con el resto de manos, pasaremos para abandonar.

River: volveremos a apostar el bote siempre que tengamos al menos una mano fuerte.

### 5.4.2.2 EP50

En este apartado se va a describir la estrategia EP50.

- **PreFlop:** en esta estrategia se va a empezar a subir más cuando haya limperts (jugadores que han igualado la ciega grande) antes de que nos llegue el turno de actuar. Empezaremos a hacer Raise Over Limperts.

También se va a empezar a aplicar la defensa de ciegas. Del mismo modo que se abrirán muchas manos para robar las ciegas, los rivales también abrirán más manos. Es decir, abrirán un rango de manos mayor. Por lo tanto, se empezará a resubir con más manos también.

A continuación, se explicará cómo jugar según las diferentes situaciones en la que se encuentre el jugador.

- Llega la mano limpia

Siempre que se juegue una mano se hará subiendo, nunca igualando la apuesta.

Si la mano llega limpia al jugador (nadie ha apostado antes que él), habría que realizar una subida a 3bb (tres veces el precio de la ciega grande), según la tabla de la estrategia en este caso.

- ROL (Raise Over Limpers)

Cuando llegue el turno de actuar y haya algún jugador que haya igualado la ciega grande, se subirá a 5 ciegas, independientemente del número de limperts(jugadores que han igualado la ciega grande), siempre y cuando la mano esté dentro del rango de reglas de la tabla en la estrategia.

- El bote llega subido

En el caso de que algún jugador iguale la ciega grande y otro suba después, independientemente de su posición, siempre se resubirá con las mismas manos que se resube cuando el rival sube desde *EP* (early position), *MP* (middle position) o *CO* (cut-off position). La apuesta se hará siguiendo la tabla de la estrategia.

- Enfrentándose a una cuarta apuesta

Cuando nos encontremos en una situación propia de la zona de robo, tendremos que ir All-in con más manos que en el nivel anterior, siguiendo la tabla de esta estrategia.

- Cuarta apuesta

Recordemos que cuando se ha subido y otro jugador resube esta apuesta, nunca se igualará dicha apuesta. Si nos encontramos en esta situación, la cuarta apuesta será siempre All-in.

Cuando se esté en una situación de zona de robo, es decir, cuando hayamos abierto desde *SB* (smallblind) o *BTN* (button) y algún rival resube, habrá que ir All-in con más manos que cuando se haya abierto fuera de la zona de robo, como muestra la tabla de la estrategia.

- **PostFlop:** en la estrategia PostFlop hablaremos de la fuerza de la mano, de los cinco tipos de manos que podemos tener: muy fuerte, fuerte, media, débil o nada.

Es muy importante tener claro qué tipo de mano se tiene según el tipo de mesa para no cometer errores importantes.

También tendremos que ajustar nuestras outs en cada tipo de mesas.

Ya no nos limitaremos a abandonar o subir frente a las apuestas del rival. Desde ahora podremos igualar. Los motivos teóricos se explican en el artículo de PME de EducaPoker<sup>12</sup>.

La fase PostFlop se divide en tres: juego cuando hemos subido preflop, cuando hemos resubidopreflop y cuando hemos pasado en la ciega grande. Dentro de cada una se analizará las fases Flop, Turn y River. Para la decisión de las apuestas en todas las fases, hay que tener en cuenta los movimientos del rival, la fuerza de la mano y el número de outs.

- Cuando se ha subido PreFlop

De nuevo, se diferencia entre botes contra un único jugador y contra varios.

Flop: jugando contra un solo rival, las apuestas serán de dos terceras partes del bote. A diferencia de cuando jugamos contra un único rival, contra varios ya no vamos a apostar en el Flop siempre.

En Turn y en River, contra un solo rival, se actuará de una forma u otra según cómo haya jugado el rival, la fuerza de la mano y el número de outs.

Contra varios rivales, si decidimos apostar en el Turn o en el River sin haber apostado el Flop, el tamaño de la apuesta será del bote. Si hemos apostado en alguna ronda anterior, nuestro tamaño será all-in.

- Juego cuando se ha pasado en la ciega grande

En PostFlop se jugará igual que como se jugó en botes que hemos subido, con una particularidad: se necesitará tener proyectos de al menos doce outs para ir All-in, ya sea subiendo una apuesta o resubiendo la subida de un rival.

- Juego cuando se ha resubido PreFlop

Cuando se resube a 9bb (nueve veces el tamaño del bote), el bote en el Flop ya es muy grande, y con una mano débil hay que querer llegar al showdown.

El tamaño de apuesta cuando se resube será de dos terceras partes del bote.

### 5.4.2.3 EP100

Seguidamente se va a describir la estrategia EP100.

- **PreFlop:** en esta estrategia se va a empezar a subir más cuando haya limpers (jugadores que han igualado la ciega grande) antes de que llegue el turno de actuar. Se empezará a hacer Raise Over Limpers (subir cuando hay jugadores que igualaron la ciega grande).

---

<sup>12</sup><http://www.educapoker.com/escuela/articulos/c7-s7-pme-per-i>

También se va a empezar a aplicar la defensa de ciegas. Del mismo modo que se abren muchas manos para robar las ciegas, los rivales también abrirán más manos. Es decir, abrirán un rango de manos mayor. Por lo tanto, empezaremos a resubir con más manos también.

Aquí se explican las formas de jugar según las diferentes situaciones:

- Llega la mano limpia

Jugamos igual que jugábamos en el nivel anterior. Si la mano llega limpia al jugador (nadie ha apostado antes que él), habría que realizar una subida a 4bb (cuatro veces el precio de la ciega grande), según la tabla de la estrategia en este caso.

- ROL (Raise Over Limpers)

Cuando el nuestro turno de actuar y haya algún jugador que haya igualado la ciega grande, se subirá a 4 ciegas, independientemente del número de limpers (jugadores que han igualado la ciega grande).

- El bote llega subido

En el caso de que algún jugador iguale la ciega grande y otro suba después, independientemente de su posición, siempre se resubirá con las mismas manos que se resube cuando el rival sube desde o *EP* (early position) , *MP* (middle position) o *CO* (cut-off position). Si según la tabla hay que hacer una apuesta, esta sería de 15bb (quince veces el precio de la ciega grande).

- Enfrentándose a una cuarta apuesta

Si se llega a la situación de cuarta apuesta, se irá All-in si la mano cumple la estrategia.

- Cuarta apuesta

Recordemos que cuando se ha subido y otro jugador nos resube, nunca igualaremos la apuesta. Si nos encontramos en esta situación, nuestra cuarta apuesta será siempre All-in (siguiendo la tabla).

- **PostFlop:** en la estrategia PostFlop hablaremos, en cuanto a la evaluación de la fuerza de la mano, de los cuatro tipos de manos que podemos tener: fuerte, media, débil o nada.

Es muy importante tener muy claro qué tipo de mano se tiene según el tipo de mesa para no cometer errores importantes.

También se tendrán que ajustar las outs en cada tipo de mesa.

Ya no habrá que limitarse a abandonar o subir frente a las apuestas del rival. Desde ahora se puede igualar. Los motivos teóricos se explican en el artículo de PME EducaPoker<sup>13</sup>.

Ahora se explicará cómo jugar en las tres fases del PostFlop. Para la decisión de las apuestas en todas las fases, hay que tener en cuenta los movimientos del rival, la fuerza de la mano y el número de outs.

- Cuando se ha subido PreFlop

De nuevo, diferenciaremos entre botes contra un único jugador y contra varios.

Flop: como sucedía en niveles anteriores, cuando se juegue contra un único rival que haya igualado la apuesta preflop, se apostará el bote el 100% de las veces. Si el rival sube la apuesta, se decidirá cómo actuar dependiendo de la fuerza de la mano y del número de Outs. A diferencia de cuando se juega contra un único rival, contra varios ya no se va a apostar en el Flop siempre.

Turn y River contra un solo rival: se actuará de una forma u otra según cómo haya jugado el rival, la fuerza de la mano y el número de Outs.

Turn y River contra varios rivales: se jugarán estas rondas de apuestas igual que las se juegan contra un único rival.

- Juego cuando se ha pasado en la ciega grande

PostFlop se jugará igual que se juega en botes que se ha subido, con una particularidad: se necesitan tener proyectos de al menos 12 Outs para ir All-in, ya sea subiendo una apuesta o resubiendo la subida de un rival.

- Juego cuando se ha resubido PreFlop

Se apostará siempre en el Flop y se acabará con All-in en todos los casos. En botes resubidos contra un solo oponente se apostará siempre la mitad del bote.

### 5.4.2.4 Configurables

Esta parte es la más novedosa e interesante del análisis de manos. La configuración de estrategias trata de, a partir de las tablas que ofrecemos en la aplicación, poder modificarlas para así tener cierta libertad para usar nuestras propias estrategias. Al existir jugadores que

---

<sup>13</sup><http://www.educapoker.com/escuela/articulos/c7-s7-pme-per-i>

no están conformes del todo con las estrategias ofrecidas, nuestra aplicación les presenta la posibilidad de modificar las tablas para poder analizar y estudiar sus propias tácticas.

### 5.4.3 Implementación

Se va a hablar ahora de la implementación de las estrategias. Básicamente se compone de cinco clases: *Estrategias.java*, *Nivel3\_EP100.java*, *Nivel3\_EP50.java*, *Nivel3\_EP35.java*, *Personalizadas.java*.

#### 5.4.3.1 Estrategias

Es la clase padre. Contiene métodos y atributos que son usados por las clases hijas, ya que son comunes a todos, como por ejemplo métodos para obtener el movimiento o la apuesta del jugador, para obtener las tablas de la estrategia, para saber si algún jugador ha apostado antes que nosotros, etc. De esta clase heredan las clases de las estrategias predefinidas, que se explican a continuación.

#### 5.4.3.2 Clases de las estrategias específicas

En estas clases están contenidas las tablas de dichas estrategias para PreFlop, así como los análisis de las manos con dichas tablas para estas estrategias. Desde estas clases, se crean objetos del tipo *NodoTabla* (que es una clase que contiene los atributos de las mismas y es usada para establecer las tablas y sus características). También está toda la lógica para analizar las jugadas posibles en la fase PostFlop. Cada una de estas clases tiene sus métodos propios para el cálculo de las jugadas en función de la estrategia.

#### 5.4.3.3 Personalizadas

Esta clase está creada para tratar las tablas que el usuario ha modificado y guardarlas para que puedan ser usadas y analizadas con las estrategias correspondientes.

Lo que cambia de las estrategias personalizadas es la tabla de estrategia, por lo que, una vez que se obtiene la tabla, se aplican los métodos de las clases anteriores.

Estas estrategias pertenecen al modelo de la aplicación, que es utilizado por el controlador de la misma para realizar el análisis de las manos con unas estrategias u otras. Tras esto, la vista muestra los resultados de estos análisis por pantalla.



## 6. PARTE SOCIAL

Como parte del trabajo, también hay que mencionar la aportación que los usuarios de la aplicación han hecho para ayudar al desarrollo de la misma, y analizar los datos recogidos.

Al terminar el desarrollo de la primera parte del análisis de estrategias (la fase de preflop), se decidió que era un buen momento para empezar a publicar la aplicación, para recibir valoraciones de usuarios y ver qué cosas se podían mejorar, corregir posibles fallos que se pudieran encontrar, y, en definitiva, ver si los usuarios la consideraban una herramienta útil que pudieran llegar a utilizar.

La primera idea fue publicarla directamente en el foro de EducaPoker, aunque en un único hilo no se podía incluir toda la información necesaria, por lo que se decidió crear una web donde describir en profundidad el propósito y el funcionamiento de la aplicación, así como alojar la misma para su descarga.

Además de por esto, el principal motivo por el que se decidió hacer la Web fue para hacer un seguimiento de la actividad de los usuarios de la aplicación, de forma que se pudiera ver el número de visitas que tiene la Web y el número de descargas se han efectuado.

La Web en cuestión es <http://pokercashanalyzer.co.nf/>.

En ella, hemos incluido tres secciones:

1. Inicio: en esta sección (figura 6.1) se explica quiénes somos y por qué estamos realizando este proyecto, se da una información básica de lo que hace la aplicación y se describe el estado actual de la aplicación y las últimas funcionalidades introducidas.  
Además, también se incluye información de contacto, por si un usuario necesita algún tipo de información adicional.



Figura 6.1 – Sección “Inicio” de la Web

2. Acerca de la aplicación: en esta sección se explica en profundidad cómo funciona el programa (figura 6.2).



Figura 6.2 – Sección “Acerca de la aplicación” de la Web

También se ilustra con figuras cómo sería una ejecución (figura 6.3).



### ¿Cómo funciona?

Al abrir la aplicación te aparecerá la siguiente ventana, en la cual podrás meter tu fichero de manos, seleccionar la casa que es y la estrategia que quieres seguir. Podrás también ver las tablas para el Preflop de la estrategia elegida.

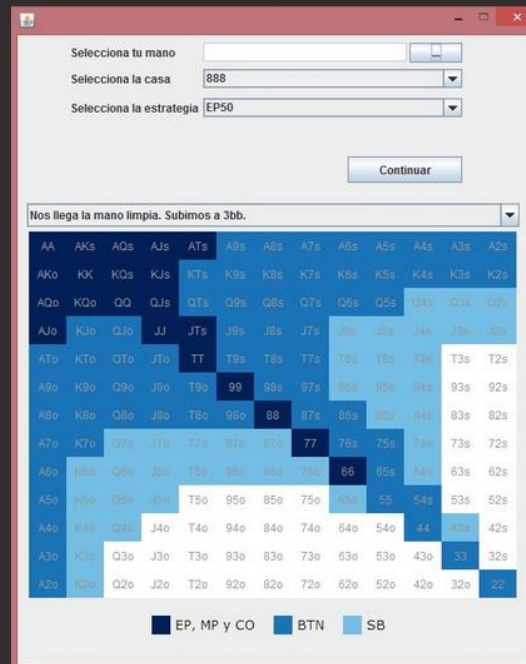


Figura 6.3 – Sección “Acerca de la aplicación” de la Web

Por último, se explica el funcionamiento de la herramienta creada para eliminar los nombres de los jugadores de las manos (figura 6.4).

### Anonimizador

1. Introducir el archivo de manos.
2. Seleccionar la ruta donde guardar el archivo generado (la carpeta de destino).

Por último, pulsar en el botón "Continuar".

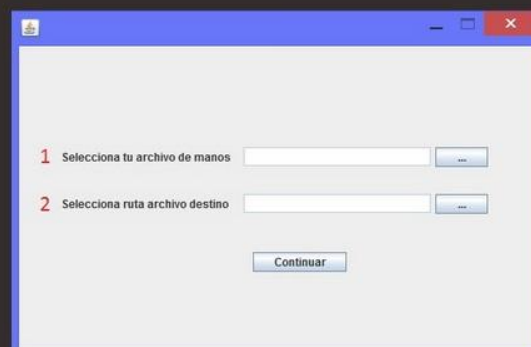


Figura 6.4 – Sección “Acerca de la aplicación” de la Web

3. Descarga: en esta sección (figura 6.5) se encuentra el enlace para la descarga de la aplicación, que es un ejecutable de java (.jar). Además, también se ha incluido una

descripción de los requisitos necesarios para el funcionamiento, así como las descargas del anonimizador de manos y una mano anonimizada por si un usuario quiere ver lo que genera esta herramienta, o bien por si quiere probar la aplicación antes de decidirse a utilizar sus propias manos.



Figura 6.5 – Sección “Descarga” de la Web

Además de la Web, se ha creado una dirección de correo electrónico para atender las posibles dudas o sugerencias de los usuarios<sup>14</sup>. También se utiliza dicho correo desde dentro de la aplicación, cuando un usuario pulse el botón de “Contactar con el desarrollador”, llegará un correo a esta dirección.

El último canal de comunicación con los usuarios, como ya se ha comentado antes, ha sido un hilo abierto en el foro de EducaPoker para dar a conocer la aplicación<sup>15</sup>.

En él, se detalla el proyecto y las razones por las que se abrió el hilo, y la respuesta de los usuarios fue bastante buena. Tanto nosotros desde nuestra cuenta como nuestro tutor hemos estado dando soporte y atendiendo a las dudas y en algunos casos a las “sospechas” de los usuarios del foro.

Cabe mencionar que estos métodos de contacto han sido muy útiles a la hora de desarrollar la aplicación, ya que ha habido usuarios que han comunicado fallos que han encontrado al probarla, o que han enviado sus manos desinteresadamente para poder hacer pruebas.

<sup>14</sup>pokerTFG@gmail.com

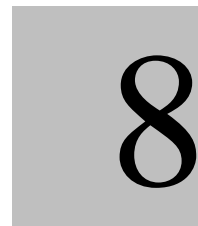
<sup>15</sup><http://www.educapoker.com/foros/general/comunidad/programas-software/analizador-manos-nlhe-siguiendo-estrategias-configurables>



## 7. GLOSARIO

- **All-in:** apostar todo. Quedarse sin fichas cuando se está apostando o viendo.
- **BB (big blind o ciega grande):** apuesta mínima que un jugador puede hacer en esa ronda, y es el doble de la ciega pequeña. La ciega grande establece, normalmente, una apuesta completa de la primera ronda. También es una posición en la mesa, dos puestos a la izquierda del repartidor.
- **BTN(button):** jugador situado en la posición de la mesa botón.
- **CO (cut-off position):** posición anterior al BTN.
- **Dealer:** repartidor. En una partida de Póker, es el jugador que realmente (o en teoría) estaría repartiendo las cartas.
- **Defensa de ciegas:** defender las ciegas es importante porque los rivales también son conscientes de que robarlas lo es, y lo van a intentar. Para defender las ciegas, consideraremos un intento de robo cuando un rival suba desde BTN o SB. Se defenderán las ciegas resubiendo un mayor número de manos preflop en estas situaciones.
- **EP (early position):** los tres asientos situados a la izquierda del repartidor.
- **EP35, EP50, EP100:** nombres de las estrategias utilizadas, donde EP viene de EducaPoker y las cifras hacen referencia al número de ciegas.
- **Flop:** segunda fase del juego. Ya hay tres cartas en la mesa.
- **Fuerza de la mano:** estadística que ofrece una idea de las opciones que se tienen de ganar y perder. Evalúa en manos muy fuerte, fuerte, media, débil o nada.
- **Mano limpia:** nadie ha realizado ninguna apuesta antes de nuestro turno.
- **MP (middle position):** posiciones intermedias, situadas después de las EP.
- **Outs:** número de cartas posibles que pueden mejorar tu mano. Mide la calidad de los proyectos de manos del jugador.
- **PME (porcentaje mínimo de éxito):** ilustra una relación entre el dinero que se arriesga en una decisión y la cantidad de veces que tiene que salir bien para que este riesgo merezca la pena. Es una forma de calcular la relación entre riesgo y beneficio, algo esencial para jugar bien al póker.

- **PostFlop:** fase posterior al preflop, que a su vez se divide en tres fases, flop, turn y river (en este orden).
- **PreFlop:** primera fase del juego. Cada jugador tiene dos cartas y no hay ninguna carta en la mesa aún.
- **Rango de manos:** conjunto de manos que puede tener un jugador en un momento determinado.
- **River:** última fase del juego. Ya están todas las cartas en la mesa: cinco.
- **ROL (Raise Over Limpers):** es la acción de subir preflop cuando un jugador o varios han igualado la ciega grande antes de nuestro turno. Es decir, somos los primeros en subir, dado que el resto de jugadores sólo igualaron.
- **SB (small blind o ciega pequeña):** es una apuesta forzosa que hace el jugador situado a la izquierda del repartidor antes de ver sus cartas. Normalmente, la ciega pequeña es uno o dos tercios de la apuesta de la primera ronda. También es una posición en la mesa.
- **Showdown:** es el momento en el que los jugadores involucrados en una mano deben enseñar las cartas para ver quién es el ganador.
- **Turn:** tercera fase del juego. En la mesa hay cuatro cartas.



## 8. CONCLUSIONES Y TRABAJO FUTURO

En esta sección se analizan las conclusiones obtenidas tras la finalización del proyecto, y también posibles vías de trabajo para una futura ampliación de la aplicación.

### 8.1 Conclusiones

Tras todo el curso académico trabajando en este proyecto, se ha implementado una aplicación que ha cumplido los objetivos previamente definidos, incluyendo tanto parsers para extraer la información de archivos que vamos a analizar, como la lógica para analizar dicha información y mostrarla de la manera más sencilla y amigable para el usuario.

La principal aportación ha sido la implementación de estrategias personalizables, ya que ninguna aplicación de este ámbito tiene esta funcionalidad.

Gracias al acceso que se ha tenido a una comunidad familiarizada con este tipo de aplicaciones, se ha podido obtener un importante *feedback* para la corrección y mejora del proyecto.

En cuanto a los objetivos cumplidos, cabe destacar el de conseguir procesar las manos que generan las casas de *PokerStars* y *888poker*. Además, se ha logrado analizar y tratar estos datos para mostrarle al usuario cómo jugar con estas estrategias. Por último, la personalización de estrategias da una libertad interesante al jugador que quiera manipular las estrategias existentes y así poder estudiar y comprobar su propia forma de jugar.

### 8.2 Trabajo futuro

Este proyecto ha dado lugar a una herramienta muy útil para el análisis del juego. Además, ofrece varias vías para extender su funcionalidad.

Una de ellas sería la inclusión de más casas de juego en los parsers. La aplicación está preparada para que sea muy sencillo añadir nuevos módulos de procesamiento de archivos de manos.

Asimismo, también sería posible la ampliación de estrategias nuevas, ya que al igual que los parsers, las estrategias también se encuentran en módulos bien diferenciados.

Por último, la aplicación está preparada para la ampliación de modalidades de juego, así como de jugadores totales en la mesa.





## 9. CONCLUSIONS AND FUTURE WORK

In this section, we analyze the conclusions obtained at the end of the project, and possible ways to expand the application in the future.

### 9.1 Conclusions

After this academic year working on this project, we have managed to implement an application which has achieved the objectives. We have developed parsers for extract the information of files, and the logic to analyze the information and show it in an easy way.

Our main contribution has been the implementation of customizable strategies, because none of the similar applications have this functionality.

Thanks to EducaPoker community access, we could have an important feedback for fixing mistakes and improve our project.

Regarding our achieved goals, we would like to mention our hand parsers, which process the *PokerStars* and *888poker* hand files. Also, we have managed to analyze this data to show the user show to play with these strategies. Finally, the custom strategies give freedom to the users for manipulating existing strategies, so they can study and check their own playstyle.

### 9.2 Future work

We think this project has led to an useful tool for analyze the game. This application offers many ways for extend its functionality.

One of this could be the integration of more poker house parsers. This application is ready for adding new parser modules in a very easy way.

Additionally, it would be posible the extension of new strategies, since they are in very different modules.

Finally, the application is prepared for the implementation of new game styles and the implementation of a different number of players.





## 10

## 10. BIBLIOGRAFÍA

- CADENHEAD, ROGERS: **Java 7**. Editorial Anaya Multimedia, 2012
- ZUKOWSKI, JOHN: **The Definitive Guide to Java Swing**. Editorial Apress, Inc., 2005
- CHACON, SCOTT: **Pro Git**. Editorial Apress, Inc., 2009
- GAUCHAT, JUAN DIEGO: **El gran libro de html5, CSS3 y Javascript**. Editorial Marcombo, 2011
- ALAN DENNIS; BARBARA, HALEY WIXOM; DAVID TEGARDEN: **Systems Analysis and Design with UML, 4th Edition**. Editorial John Wiley & Sons, 2012
- FLYNN, MATT: **Professional No-LimitHold 'em: Volume I: 1**. EditorialTwo Plus Two Pub, 2007.
- MILLER, ED: **Small Stakes No-LimitHold'em**. Editorial DimatEnterprises, Inc., 2010
- SKLANSKY, DAVID: **Ganar al poker**. EditorialLa Esfera De Los Libros, S.L., 2009
- Escuela de póker (último acceso: 06/06/2014): <http://www.educapoker.com/escuela>
- Definiciones básicas (último acceso: 06/06/2014): <http://www.pokerstars.es/poker/terms/>
- Uso de JavaMail (último acceso: 06/06/2014):  
<http://www.chuidiang.com/java/herramientas/javamail/empezar-javamail.php>
- Uso de paneles extensibles en Java (último acceso: 06/06/2014):  
<http://www.coderanch.com/t/341737/GUI/java/Expand-Collapse-Panels>